FINAL PROJECT STATUS REPORT
By Connor Levesque & Max Shashoua

Title:
Othello - A Love Story

Members:
Connor Levesque
Max Shashoua

Summary:
We have created a comprehensive list of everything we will have to implement. We have written a framework to store board states using pytorch tensors, and have written the functions to find legal moves and identify isomorphic board states (reflections or rotations of a state). We have also sketched out the Monte Carlo Tree data structure we will implement. Our next steps are to implement the Monte Carlo Tree Search using that data structure, build a game automation class that works first with a random mover and then integrates the Monte Carlo tree search. Once that works, we will plug in the neural net that we will implement using pytorch. We will model our net's architecture roughly on the one used in AlphaGoZero. Finally, we will automate the testing and self play, train, then test against a random-mover, then minimax, then alpha-beta search player.

Our self-play training algorithm is as follows:

A move is made using data that accumulate during k iterations of MTCS. MTCS works by searching through a tree of game states until a leaf is hit. At the leaf, the algorithm expands that leaf and evaluates the next moves using the net, then backpropagates the evaluation back up the tree to the root node (i.e. the current state). Notice that each iteration adds to the tree, thus expanding the search's vision of the game.

The search chooses its path using a search criterion that comes from the data we collect from the search and evaluations, which tries to find a balance between looking at new paths (expanding shallower leaves) and searching deeper. This part uses the evaluation output of the net. At the end of the k MCTS iterations, we are left with estimations of move quality data from the searches which we use to pick the best move.

Once we make this move, we will repeat this algorithm for the next game state, only this time, we have a small tree working to start with from the last set of MCTS iterations from the last move. Once the game is complete, we are able to associate each board state and move with a win or a loss. We will train the move predictions part of the net on the moves we made in the game, and the evaluation of board states part on the win/loss label we can now attribute to each state since the game is over.

Results:
We do not have any numerical results yet.

Problems:
We have not hit any problems so far. The tough part has been understanding the paper on AlphaGoZero well enough to start implementing and designing. However, we are confident we have a good handle on the methods now.

Hours:
Connor - 9h
Max - 6h

Code:
See attached .zip file